Frame Gallery Naming Conventions

This section provides information on naming conventions within frame gallery.

- Reserved Identifiers for Key Values
- Conventions for Message Text
- Natural Object Names
- Frame Gallery Object Names

Reserved Identifiers for Key Values

To distinguish frame gallery objects from those which are created during the development of an application, certain prefixes and value ranges are reserved for frame gallery objects.

The identifiers listed in the table below must not be used for any of the following:

- Names of objects of the documentation tool being used;
- Names of all Natural modules;
- Designation of variables and database fields;
- Key IDs for:
  - applications;
  - commands;
  - object types;
  - functions;
  - function groups;
  - background procedures;
  - dialog types;
  - symbol bars.

The reserved identifiers are:

Identifier	Object
Z*	Natural module names
GZ_*	Global data contained in global data areas
LZ_*	Local data contained in local data areas
PZ_*	Parameter data in parameter data areas
#Z	Local data
Z_*	Applications, commands, object types, function groups, background procedures, dialog types, tool bars
Z*	Functions

Conventions for Message Text

The application shell requires message texts. These texts are partly referenced from the frames. The following numbering scheme is used:

- 0001-0999: system messages
- 1000-9999: application messages

The application shell messages are in the file msg1.dat (English) and msg2.dat (German) respectively. These files serve as the basis for the generation of a Natural message file.

System Messages

This message area contains

- General error messages such as "data record not found";
- Error messages for the application shell.

You should not make additional entries in this message area.

Application Messages

This message area is reserved for messages used by applications implemented using the application shell.

It is recommended to classify messages for various applications by assigning specific ranges to each application. This avoids possible conflicts in message number usage among applications.

Natural Object Names

This section describes the naming conventions for Natural objects as well as the designation of program variables and database fields. These naming conventions are general recommendations for the designation of individual objects.

Structure

Natural objects which do not contain any language-dependent components must be named according to the following structure:

BOOXXXXN

where *B* is the business area abbreviation (for example, P for Purchasing), *OO* is the object type abbreviation, *XXXX* is the function identifier and *N* is the Natural object type.

Natural objects which are language-dependent must be named according to the following structure:

BOOXXXLN

where *B* is the business area abbreviation, *OO* is the object type abbreviation, *XXX* is the function identifier, *L* is the language code and *N* is the Natural object type.

Business Area Abbreviation

The abbreviation of the business area to which the function belongs. Such abbreviations must be set organization-wide to ensure uniqueness and consistency.

Examples:

P	Purchasing
S	Sales
I	Inventory

Object Type Abbreviation

The abbreviation of the object type to be processed.

If the Natural module is used together with a specific information object, the abbreviation of the information object must be as descriptive as possible.

Example:

CU	for Customer

If the Natural module is used for a specific central task, the first position of the object type abbreviation must contain the character (X) and the second position must contain an abbreviation of the processing task.

Example:

XD	for Dating

If the Natural module is of primary importance to the entire application and can be used throughout the application, the object type abbreviation must be set to XX.

Function Identification

An abbreviation which identifies the function uniquely. If the module is a central, frequently reused module, it is important to use an abbreviation which is as descriptive as possible.

If the module is used only for a specific information object or processing task, the following guidelines can be used:

Positions 1 and 2	Type of Processing
AM	Multiple record access module
AS	Single record access module
BR	Listing of objects for an object type (browse dialog)
DD	Delete (deletion subprogram)
KD	Key ID for dialog (Key dialog)
MD	Maintain processing function (maintain dialog)
MR	Multiple records (mass processing dialog)
MW	Modal window processing (modal window)
SU	Lower level window (subdialog)
NS	Individual dialog (nonstandard dialog)

Positions 3 (and possibly 4) are used as additional numeric qualifiers (01-99) in the event that there is more than one module per type of processing. Otherwise, these positions may be set to zero.

Natural Language Code

The code of the language belonging to the interface of the Natural module.

If the system is multilingual, you must specify the special character which is to replace the current language code at runtime.

Natural Object Type

The Natural object type. Examples are shown below.

- A = parameter
- C = copycode
- D = dialog
- G = global data area
- L = local data area
- N = subprogram
- P = program
- S = external subroutine
- T = text

Please note the following difference in dialog designations between Natural and frame gallery:

- In Natural, a dialog is designated by Natural object type ’3’.
- In frame gallery, a dialog is designated by ’D’.

Examples:

TXXINF1D

T	Business area: T for Travel Cost Reimbursement
XX	Internal, reusable module
INF	Information display
1	Language code: 1 for English
D	Natural object type: D for dialog

PCUMD02D

P	Business area: P for Purchasing
CU	Object type abbreviation: CU for Customer
MD	Function abbreviation: MD for main dialog
0	Indicates that this is the only such module for this area
2	Language code: 2 for German
D	Natural object type: D for dialog

SORAS00N

P	Business area: S for Sales
OR	Object type abbreviation: OR for Order
AS	Function abbreviation: AS for access single record
0	Indicates that this is the only such module for this area
0	Language code: 0 indicates no language dependency
N	Natural object type: N for subprogram

Field Names

Variable Names

The general structure used for variable names is as follows:

prefix_ BOO _name _suffix

prefix	G_ global variable
	L_ local variable defined in an local data area
	P_ parameter data defined in a parameter data area
	# user-defined local variables
B	Business area in which variable is used. May be omitted if the variable is not to be associated with a specific business area.
OO	Object type for which variable is used. May be omitted if the variable is not to be associated with a specific information object.
name	The name of the variable. The name must be as descriptive as possible.
suffix	Further classification of the variable. **Examples:** _CV Control variable _FROM Variable used for starting value _THRU Variable used for ending value _A20 Variable used for field format

Handle Names

In event-driven applications, handle names are used to describe dialog elements. In order to be able to provide a clearer description of a business element used within a dialog, the following naming structure is recommended:

prefix _ control-abbreviation _BOO _ name

prefix	Variable type. In that handles are defined locally in a dialog, the character # is used here. Frame gallery-specific handles are identified with the characters #Z.
control-abbreviation	Natural-defined abbreviation for a control variable. **Examples:** **BM** - bitmap **PB** - push button **IF** - input field
B	Business area in which variable is used.
OO	Object type for which variable is used.
name	Descriptive name of the handle.

Frame Gallery Object Names

These naming conventions are general recommendations for the designation of individual objects.

General

The key ID for the application shell administration system is generally preceded with the default abbreviation Z. Therefore, user data records must not begin with Z. For further information, see Reserved Identifiers for Key Values.

Frame Gallery Function Names

In order to clearly associate a function with an application via the function key ID, the business area and object type abbreviations must precede the key ID.

Examples:

ZUS_MNT

Z	Business area abbreviation reserved for frame gallery
US	Object type abbreviation (user maintenance)
MNT	Key ID for maintenance function

SCU_DIS

S	Business area abbreviation (S for Sales)
CU	Object type abbreviation (CU for Customer)
DISP	Key ID for function which displays a data record